# A Parallel Tabu Search for Conformational Energy Optimization of Oligopeptides

**L. B. MORALES,[1] R. GARDUÑO–JUÁREZ,[2] J. M. AGUILAR–ALVARADO,[3] F. J. RIVEROS–CASTRO[3]**

[1]*IIMAS-UNAM, Apdo. Postal 70–221, 04510, México D.F.*
[2]*CCF-UNAM, Apdo. Postal 48–3, 62250 Cuernavaca, Morelos, México*
[3]*FC-UNAM, Circuito Exterior, Ciudad Universitaria, Coyoacan, 04510, México D.F.*

*Received 8 June 1998; accepted 16 September 1999*

**ABSTRACT:** We have developed and implemented a tabu search heuristic (TS) to determine the best energy minimum for oligopeptides. Our test molecule was Met-enkephalin, a pentapetide that over the years has been used as a validation model for many global optimizers. The test potential energy function was ECEPP/3. Our tabu search implementation is based on assigning integer values to the variables to be optimized, and in facilitating the diversification and intensification of the search. The final output from the TS is treated with a local optimizer, and our best result competes both in quality and CPU time with those reported in the literature. The results indicate that TS is an efficient algorithm for conformational searches. We present a parallel TS version along with experimental results that show that this algorithm allows significant increases in speed. © 2000 John Wiley & Sons, Inc. J Comput Chem 21: 147–156, 2000

**Keywords:** protein structure prediction; heuristic algorithms; protein folding; conformational space search; tabu search

*Correspondence to:* R. Garduño–Juárez; e-mail: ramon@ce.fis.unam.mx

Contract/grant sponsor: PAPIIT-DGAPA-UNAM; contract/grant number: IN-102397

Contract/grant sponsor: CRAY-UNAM Research Fund; contract/grant number: SC-002195

Contract/grant sponsor: CONACyT; contract/grant number: 25245-E



CCC 0192-8651 / 00 / 020147-10

## Introduction

Peptides are short polymers made up of a few to a few tens of amino acids. Many of these have meaningful roles in biochemistry and biophysics. Some sequences of peptides have a clear tendency to form well-defined three-dimensional structures, that is, to fold. Peptides are also useful as model systems for much larger peptide chains known as proteins. The naturally occurring three-dimensional structure of a protein, its "tertiary structure," is believed to be uniquely determined by its "primary structure," the sequence of amino acids of which the protein is composed. Anfisen[1] in his "thermodynamic hypothesis" proposes that the native state of a protein is the structure that minimizes the free energy. By definition, such a state would be at the global minimum of free energy relative to all other states accessible on that time scale. Thus, the conformational search, or folding, can be posed as an optimization problem.

Conformational search of peptide molecules, to a first approximation, can be thought of as the problem of finding the 3D molecular structure that corresponds to the lowest local minimum of an appropriate mathematical function describing the potential energy of the system. Computer simulations are often used to carry out this task. A major concern in computer simulations is to obtain a set of low-energy conformations with biological significance; that is, finding those conformations that are near the thermodynamic native state.

Folding a protein from only a knowledge of its amino acid sequence is a formidable task. Because it is computationally impossible to test all possible conformations to determine the global minimum, it is necessary to develop methods that can land upon a global minimum without testing all conformational possibilities. This is a challenging optimization (minimization) task. In many cases the detailed properties of the potential function to be minimized are not known. Even if the function is differentiable, one can often encounter nonconvex surfaces, and the local properties of the function can be different in the different search regions, i.e., the basins can have different size or depth, the smoothness can vary, etc.

Many different force fields for proteins have been designed as a summation of a set of potential energy contributions. Among the most used ones are: ECEPP,[2] MM2,[3] ECEPP/2,[4] CHARMM,[5] DISCOVER,[6] AMBER,[7] GROMOS87,[8] MM3,[9] and ECEPP/3.[10] Most of these have a large number of local minima. In general, protein folding with any force field is a NP-hard problem,[11] where the time needed to locate the lowest minimum grows exponentially when the number of variables grows linearly. A major challenge in this type of global optimization problems is that there is no clear mathematical basis for efficiently reaching the global minimum, thus finding the latter in an accurate and speedy way is of general interest.

To reduce the size of the problem one takes advantage of the fact that under biological conditions some internal motions of protein molecules occur on a time scale much smaller than others. Experimentally, the average values of covalent bond distances and covalent bond angles are fairly constant, and lead to the assumption that conformational changes observed in the dihedral angles could fully determine the overall shape of the protein molecule. Thus, if one specifies the position of all atoms in the protein molecule as a function of its internal coordinates, under the assumption of constant bond lengths and bond angles, the problem drastically reduces the number of its degrees of freedom.

Although the size of the problem can be reduced when the energy function is written in terms of torsional angles, it is known that in this form the energy function is no longer partially separable, meaning that it is no longer much less expensive to reevaluate the energy if only a few variables change than if they all change. To overcome this effect, a number of workers have devised interesting stochastic and nonstochastic methods, which impose constrains and bias the search towards the region where the global minimum could be found. Among stochastic methods employed to predict oligopeptide 3D structures are Monte Carlo with minimization[12a, 12b] (MCM), simulated annealing[13] (SA), threshold accepting[14] (TA), free energy Monte Carlo with minimization[15] (FMCM), multicanonical ensemble[16] (ME), conformational space annealing[17] (CSA), and genetic algorithms[18] (GA). Among nonstochastic methods we find molecular dynamics with minimization[19] (MDM), dynamic programming[20] (DP), the diffusion equation method[21a] (DEM), the mean-field technique[22] (MFT), and a global optimization procedure known as $\alpha$BB.[23]

In this article we take an approach to minimize the ECEPP/3[10] energy function based on tabu search[24] (TS), a stochastic optimizer developed to treat complex combinatorial optimization tasks. Tabu search has the advantage that only function values are used, differentiability and continuity are not required, and it is characterized by the use

of "memories" during the search. In operation research literature, tabu search has proven to be better than simulated annealing, both in the CPU time required and in the quality of the solutions found for many complex problems. Our test molecule is that of Met-enkephalin, a pentapetide that has been used as a validation model for many global optimizers, and because its lowest energy conformation for the potential energy function ECEPP/3 is known.[23] We first present the problem we are dealing with in a mathematical fashion, then we discuss the general principle of the tabu search heuristic and explain how to use tabu search for conformational search. Finally, we present our computational results and compare our best results with those reported by other authors who have employed methods different from TS. We propose the use of a version of TS in parallel to improve the CPU time needed to find the lowest energy structure. Such a concurrent version was developed for parallel computing on a SGI ORIGIN 2000 computer with close to ideal speed up.

## The Problem

As indicated above, the conformation of a protein with a sequence of $N_{res}$ amino acid residues in the peptide chain can be described by a set of dihedral angles $\phi_i$, $\psi_i$, $\omega_i$, where $i = 1, \ldots, N_{res}$ on the backbone, plus a set of dihedral angles $\chi_i^j$, $i = 1, \ldots, N_{res}$, $j = 1, \ldots, J^i$, where $J^i$ denotes the dihedral angles of the side group on the $i$th residue. If one wishes to allow capping of the peptide, then one has to include two more sets of dihedral angles. One could be defined as $\varphi_k^N$, $k = 1, \ldots, K^N$ for those dihedral angles on the amino end group, and the other could be defined as $\varphi_k^C$, $k = 1, \ldots, K^C$ for those dihedral angles on the carbonyl end group.

In this report the complete ECEPP/3[10] force field was used. This force field is built upon the assumptions that the bond lengths and angles are at their equilibrium values, and that the resulting function is in reality a conformational energy surface made of a summation over interactions of types 1–4 and higher. These interactions take into account electrostatic, nonbonded, hydrogen bond, and torsional energies, plus other empirical terms that take into account a loop closing potential in the case that the peptide has intramolecular disulfide bonds, and fixed conformational energies for the propyl and hydroxypropyl residues. A condensed description of the ECEPP/3 force field could be written as:

$$U = U_{\text{elec}} + U_{\text{nonb}} + U_{\text{hb}} + U_{\text{tor}} + U_{\text{loop}} + U_{\text{S-S}}$$

where

$$U_{\text{elec}} = \sum_i \sum_{i \neq j} 332.0 q_i q_j / D r_{ij}$$

$$U_{\text{nonb}} = \sum_i \sum_{i \neq j} FA / r_{ij}^{12} - C / r_{ij}^{6}$$

$$U_{\text{hb}} = \sum_h \sum_x A'_{hx} / r_{hx}^{12} - B_{hx} / r_{hx}^{10}$$

$$U_{\text{tor}} = \sum_k (U_0 / 2.0)(1 \pm \cos n_k \theta_k)$$

$$U_{\text{loop}} = \sum_l B_l \sum_{i=1}^{i=3} (r_{il} - r_{i0})^2$$

$$U_{\text{S-S}} = \sum_s A_s (r_{4s} - r_{40})^2$$

All constants are estimated by fitting of experimental data.[10, 25]

Given these definitions, the potential energy minimization problem can be summarized as follows:

$$\text{minimize } U(\phi_i, \psi_i, \omega_i, \chi_i^j, \varphi_k^N, \varphi_k^C)$$

subject to the particular constrains:

$$-180^\circ \leq \phi_i, \psi_i < +180^\circ \qquad i = 1, \ldots, N_{res}$$

$$-10^\circ \leq (\omega_i - 180^\circ) \leq +10^\circ \qquad i = 1, \ldots, N_{res}$$

$$-180^\circ \leq \chi_i^j < +180^\circ \qquad i = 1, \ldots, N_{res},\ j = 1, \ldots, J^i$$

$$-180^\circ \leq \varphi_k^N, \varphi_k^C < +180^\circ \qquad k = 1, \ldots, K^N,\ k = 1, \ldots, K^C$$

We also have included anticorrelations in angles $\phi_i$, $\psi_{i+1}$ as defined in a previous article[13] to further reduce the space search.

## Tabu Search

The tabu search heuristic is a search procedure for solving complex combinatorial optimization problems of a general type. It has been applied successfully to vehicle routing,[26] large traveling salesman,[27] job shop scheduling,[28] a bridge club scheduling,[29] evaluation of chemical distance,[30] and protein conformation on a lattice model.[31] This procedure has also been extended to optimization of continuous-valued functions,[32, 33] one of which has been applied to molecular docking[34] of small molecules.

Generally speaking, TS can be designed to perform the following task: minimize $f(x)$, subject to $x \in X$, where $f$ is a cost function, and $X$ is a set of feasible solutions.

Tabu search is an iterative process. It starts from an initial feasible solution and tries to reach a global minimum by moving from one solution to another. To accomplish this, we must define a set $M$ of simple modifications that can be applied to a given solution to move to another solution. These modifications are called moves. The notation $x' = m(x)$, $m \in M$ indicates that $m$ transforms $x$ into $x'$. This leads us to the definition of neighborhood $N(x)$, an ingredient common to most heuristic and algorithmic procedures for optimization. For each feasible solution $x$, the neighborhood $N(x)$ is the set of all feasible solutions directly reachable from $x$ by a simple move $m$ in $M$. At each step of the iterative process, we generate a subset $V^*$ with $j$ elements, and we move from $x$ to the best solution $x^*$ in $V^*$, whether or not $f(x^*)$ is better than $f(x)$. If $N(x)$ is not large, it is possible to take $V^* = N(x)$. The method of examining the entire neighborhood $N(x)$ is best for writing a parallel tabu code because it allows a confortable balancing of the work load between the processors. To reduce the sampling size of $V^*$ one can take the first move that improves the current solution (however, if there is no move that improves the current solution, then one has to examine all neighbors in $V^*$); in this way one can speed up the search because the mean calculation time of a step is less than the one needed in the previous method; however, the evaluation time is not constant, and the steps taken are not as good.

Up to this point, the algorithm is close to a local improvement technique, except that we may move from $x$ to a worse solution $x^*$, and, thus, we may escape from any local minima in $f$. To prevent cycling, a queue called the tabu list $T$ of length $|T| = t$ is provided. Its aim is to forbid moves between solutions that reinstate certain attributes of past solutions. After $t$ iterations they are removed from the list and free to be instated again. The tabu list is also called short-term memory, because it stores information about the $t$ most recent moves.

In many TS implementations the short-term memory is complemented with a long-term memory, whose purpose is to diversify the search and to move to unexplored regions; its function is usually based on a frequency criterion. Other rules of diversification and intensification have been proposed in the literature to improve the search.[35, 36]

Unfortunately, the tabu list may forbid certain interesting moves, such as those that will lead to a better solution than the best found so far. To cancel the tabu status of a move when this move is judged useful an aspiration criterion is also introduced

Stopping rules must also be defined. In many cases a lower bound $f^*$ of $f$ is known in advance. As soon as we are close enough or we have reached this bound we may interrupt the algorithm. In general $f^*$ is not available with sufficient accuracy, as it is the case in this study; thus, the stop criterion is met whether a fixed maximum number of iterations is reached, or if a given maximum number of iterations have been performed without improving the best solution obtained so far.

## Adaptation of Tabu Search

The most important points in the implementation of a general TS to our particular application are: the search space $X$, the cost function $f$, the neighborhood $N(x)$, the method of choosing an initial solution, the length of the tabu list $T$, the aspiration criterion and the stop criterion.

Let $k$ denote the number of amino acid residues in the molecule. Let $\phi_i$, $\psi_i$, and $\omega_i$ denote the dihedral angles in the skeleton corresponding to the $i$th amino acid residue, where $i = 1, \ldots, k$, and let $\chi_1, \ldots, \chi_m$ be the dihedral angles in the lateral chains. If $\Omega$ denotes a set of integer angle values between $-180°$ to $+180°$, then any vector $x = (\theta_1, \ldots, \theta_n) = (\phi_1\psi_1\omega_1, \ldots, \phi_k\psi_k\omega_k, \chi_1, \ldots, \chi_m) \in \Omega^{3k+m}$ determines a three-dimensional conformation of the molecule, and $n$ is the total number of variables (in this case it is equal to $3k + m$). For the intensification process, *vide infra*, the set contained in are multiples of 0.5 degrees.

Now we define the space search as:

$$X = \{x = (\theta_1, \ldots, \theta_n) \in \Omega^{3k+m} \mid 180 - \sigma_0 \le \omega_i \le 180 + \sigma_0 \text{ and } -\sigma_1 \le \phi_i + \psi_{i+1} \le \sigma_1,\ i = 1, \ldots, k\},$$

$\sigma_0$ and $\sigma_1$ have been defined previously in an earlier article from this group.[13] It is important to stress that the anticorrelation $\sigma_1$ appears to be important in regions where side chains are close in space and in loops with reverse turns.[37, 38]

The cost function $f(x)$ is the empirical energy function ECEPP/3, which is designed to work in angle space $X$, while keeping bond length and bond angle values constant, and where no solvent effects are included.

We define a single move $m_s$ for $\theta$ belonging to the set of non anticorrelated angles, as the vector $m = (i, \theta_i, \theta_i')$ that transforms $x = (\theta_1, \ldots, \theta_i, \ldots, \theta_n)$ to $x' = (\theta_1, \ldots, \theta_i', \ldots, \theta_n)$ with $1 \le i \le n$, $-180 \le \theta_i \ne \theta_i' \le 180$, and $|\theta_i - \theta_i'| \ge 2$; for $\omega$ angles, we take $180 - \sigma_0 \le \theta_i \ne \theta_i' \le 180 + \sigma_0$. To take into account the anticorrelations, it is necessary to define

the composite moves. Thus, if $m_1 = (i, \phi_i, \phi_i')$, and $m_2 = (i+1, \psi_{i+1}, \psi_{i+1}')$, such that $-\sigma_1 \le \phi_i + \psi_{i+1} \le \sigma_1$, and $-\sigma_1 \le \phi_i' + \psi_{i+1}' \le \sigma_1$, the composite move of $m_1$ and $m_2$ transforms $x = (\theta_1, \ldots, \theta_i, \ldots, \theta_{i+4}, \ldots, \theta_n)$ into $x' = (\theta_1, \ldots, \theta_i', \ldots, \theta_{i+4}', \ldots, \theta_n)$. According to these definitions, the number of neighbors for an $\omega$ angle are 20, otherwise they are 359. In each iteration we generate randomly a subset of $V^*$ with 5% of $N(x)$. Therefore, the maximum number of moves (angles) tested in each iteration is 346. For the sequential version of tabu search we implemented a method were one takes the first move that improves the current solution; thus, the maximum number of moves tested is much less than 346; it can be as small as 1. For the parallel version of tabu search we take the best solution after exploring all neighbors contained in $V^*$.

The tabu list is constructed as follows. When a single move $m = (i, \theta_i, \theta_i')$ is performed, we forbid any move $m = (i, \theta_i', \theta_i'')$ during $t$ iterations if $\theta_i'' \in [\theta_i' - d, \theta_i' + d]$. When the composite move of $m_1 = (i, \theta_i, \theta_i')$ and $m_2 = (j, \theta_j, \theta_j')$ is performed, any move $m_h = (h, \theta_h', \theta_h'')$ is declared forbidden during $t$ iterations if $\theta_h'' \in [\theta_h' - d, \theta_h' + d]$ for $h = i$ or $h = j$. The values of $t$ and $d$ (tabu distance) are adjusted experimentally. At the beginning of the process the tabu list is empty.

In this work we take the aspiration criterion as a function $A(f(x))$ defined for every value of $x$. This criterion allows the tabu status of a move from $x$ to $x'$ to be overridden if the value $f(x')$ is strictly better than the best value obtained so far. This means that for any solution $x$, $A(f(x)) = f(x^0)$, where $x^0$ is the best solution found so far, the tabu status of a move from $x$ to $x'$ may be dropped if $f(x') < A(f(x^0))$. Initially we set $A(f(x)) = \infty$.

To improve the search, our tabu implementation also includes other components such as a long-term memory, a diversification process, and an intensification process.

We make use of a long-term memory in the form of a frequency vector $(b_i)$ of dimension $n$. At the beginning of the procedure the frequency vector has zeros. When the move $m = (i, \theta_i, \theta_i')$ is performed, the corresponding counter in the frequency vector is changed to $b_i = b_i + 1$, and the values of nonimproving moves in the cost function through changes in the value of the dihedral angle $\theta_i$ are increased by $b_i$.

Our algorithm consists of three phases that are iteratively executed. In the first phase the moves are selected according to the short memory, the aspiration criterion, and the frequency vector. In a second phase called diversification, TS takes again the initial random solution of the first phase, and tries to avoid the search in regions similar to those visited already. This is achieved by storing all accepted moves in the first phase in a vector $V = (v_1, v_2, \ldots, v_k, \ldots, v_l)$. When the move $m = (i, \theta_i, \theta_i')$ is performed in the $k$th iteration, we put $v_k = i$. Then, in the $k$th iteration all moves that involve the $v_k$th dihedral angle are prohibited. At the beginning of this phase the tabu list, the aspiration criterion, and the frequency vector are initialized. In the third phase, we take the best solution found so far, and use it to start a process called intensification. In this phase, the search continues in regions where better solutions may be found by taking moves $m = (i, \theta_i, \theta_i')$ where $\theta_i' = \theta_i + 0.5^*\text{random}[-20, +20]$ and $|\theta_i - \theta_i'| > 1$. The length of tabu list $t$, and the tabu distance $d$, are now reduced to half of their original size. Also at the beginning of this phase the tabu list, the aspiration criterion, and the frequency vector are initialized.

The last concept to explain is the stopping criterion used. This was triggered when the number of iterations without improving the best solution is greater than a nimax limit.

## Parallelization of Tabu Search

Any iterative local search method is often plagued by large CPU time to obtain good solutions. This is generally due to a large number of iterations or to intensive computation iterations. However, if the search could be optimized on a single processor computer, then it is possible to make use of a multiprocessor system to accelerate the search. To increase the number of iterations per time unit we can do one of three things: (a) accelerate the calculations within each iteration, (b) execute several moves simultaneously, and (c) execute several independent searches.

The first choice implies making the evaluation of the cost function, the move values, or even the choice of the best move in parallel. The second one leads to a problem of decomposition or duplication, meaning that if several moves may be performed simultaneously, then they are independent and the problem can be decomposed. Within this parallelization one could envision a decomposition where move evaluation is a costly sequential procedure, but it is possible to concurrently execute the election of the best neighbor. This is the approach used here because in TS the component that consumes the most CPU time is the neighborhood evaluation from the currently available solutions. Thus, the absolute TS speed can be improved if it is written for parallel execution. In a

serial version of TS, each neighbor is evaluated sequentially, and a new neighbor is not generated until the previous one is done. In a parallel version of TS multiple neighbors can be evaluated concurrently, taking advantage of the fact that every neighbor is independent of the rest. In fact, it has already been suggested[39] that the most efficient way of executing a tabu search concurrently is based on distributing across many processors the most computationally intensive phases of the algorithm. This is partially achieved by allowing the search within a fixed number of neighbors or within the entire neighborhood.

Our algorithm was implemented by partitioning the set of possible movements on $\rho$ subsets of approximately the same size, and evaluating every partition in $\rho$ different processors. In this way every processor finds its best move. If one processor is chosen, for example processor zero, this processor receives the $\rho - 1$ best moves proposed by the other processors, and chooses the best move among these $\rho$ moves including its own. Later, this processor communicates the chosen move to the other processors, which allows all processors to perform the best move and update the tabu list, the frequency vector, and the aspiration criterion. The network configuration used for the parallel TS is a tree structure in which each nonzero processor (node) is connected to the zero node (see Fig. 1).

This technique requires extensive communication since two steps of synchronization between processor zero and each other processor are unavoidable for each iteration. Nevertheless, because we assigned about the same work load to each process, we can expect them to take approximately the same amount of time between synchronization. Moreover, the communication time to send and receive the operation is significantly smaller than the work load that is actually performed concurrently between the synchronizations.

The parallelization experiments were performed in a SGI ORIGIN 2000 with 32 processors operating under IRIX 6.4, physically a distributed memory computer, but logically a shared memory computer. All programs were written in standard C language, and we used MPI,[40,41] a multiple passing scheme for interprocessor communication. The ECEPP/3 FORTRAN source code was translated into C using the f2c software.[42] The resulting code was modified where needed in order to accommodate its use in the parallel version of TS.

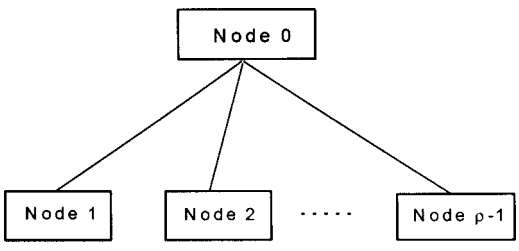


**FIGURE 1.** Network configuration for the parallel TS algorithm.

## Results and Discussion

Met-enkephalin has 24 dihedral angles, that according to our definition of a space search means that a set of 24 variables will be optimized. The best values for the parameters defined in the Adaptation of TS section were found experimentally; these were $nimax = 200$, $d = 5°$, and $t$ an integer between 40 to 60. It is obvious that to further improve the performance of the heuristic, one has to experiment with different values of its parameters. Through the remainder of this article the term minimization implies the application of a local minimizer (SUMSL[43] in particular) to a given conformation of Met-enkephalin.

It is of interest to cite the computer time for one processor of a SGI Origin 2000 computer, for which our TS procedure was compiled with the option −O2. One hundred independent runs with random initial solutions were carried out. In Table I we present the statistics of two sets of 50 runs each, the first with parameters $nimax = 200$, $d = 5°$, and $t = 40$, and the second with parameters $nimax = 200$, $d = 5°$, and $t = 60$. In all cases the energy function was evaluated 170,000 times in average for a complete run. This corresponds to about 260 s of CPU time on the SGI Origin 2000. In some cases the lowest energy minimum was found within the first or second phases of our TS implementation. Some of the low-energy structures obtained led to other structures near the global minimum of Met-enkephalin after their minimization. In these experiments we had a turn out of about 50% for Met-enkephalin structures with conformational energies lower than −9.0 kcal/mole (see Table I).

In Table II we compare our best TS result with the best low-energy conformations of Met-enkephalin reported using other global optimization methods. The first three columns show the dihedral angles of the lowest energy structure for Met-enkephalin obtained with the methods known as MCM,[15a] BB,[23] and CSA,[17], respectively. The fourth column shows the dihedral angles for the best TS conformation

**TABLE I.**
**Statistics for 100 Independent Runs of TS.**

| Energy | $t = 40$; $d = 5$[a] | | $t = 60$; $d = 5$[a] | |
|---|---|---|---|---|
| Rank | TS Results | TS + SUMSL | TS Results | TS + SUMSL |
| −1 | 1 | — | — | — |
| −2 | — | — | — | — |
| −3 | 7 | 3 | 2 | 1 |
| −4 | 10 | 4 | 8 | 3 |
| −5 | 14 | 7 | 13 | 1 |
| −6 | 18 | 5 | 12 | 10 |
| −7 | 18 | 17 | 23 | 12 |
| −8 | 17 | 26 | 26 | 17 |
| −9 | 10 | 8 | 12 | 9 |
| −10 | 2 | 27 | 2 | 39 |
| −11 | 2 | 2 | — | 3 |
| −12 | — | 4 | — | 3 |

[a] See text for details.

within the ECEPP/3 potential function. The fifth column lists the dihedral angles obtained after minimization of the structure presented in column four. The sixth column lists the dihedral angles we obtained after minimization of the lowest energy structure obtained with the TA[14] algorithm and ECEPP/2. The seventh column lists the dihedral angles obtained after minimization of the lowest energy structure reported by Wang and Pachter,[44] who employed a simulated annealing variant (ASA) and ECEPP/2. The energy values listed for MCM, BB, and CSA come from a single-point calculation with the most recent version of ECEPP/3[10] upon the structures reported. This was done because we found a few discrepancies in the energy value reported in those articles. For instance, Nayeem et al.[15a] reported a value of −12.904 kcal/mol for an ECEPP/2 energy; Androulakis et al.[23] and Lee et al.[17] both reported an energy value of −11.707 kcal/mol for their "global" ECEPP/3 energy structures. These values are clearly an artifact arising from the slight differences in the ECEPP parameters used in the computations cited.

Despite the fact that we used an integer approximation for all dihedral angles on Met-enkephalin, the best minimum energy conformation obtained from our TS is sufficiently close to those reported by other authors. To have a measure of how well this structure agrees with those obtained with the BB and CSA procedures, we have calculated its root mean squares. Thus, Arms is the rms relative to Androulakis et al.[23] and Lrms is the rms relative to Lee et al.[17] We can see that the precision achieved with TS using discrete variables (fourth column in Table II) is in good agreement with the values of Androulakis et al.[23] (Arms = 3.96). If we minimize this structure, the resulting dihedral angles are in excellent agreement with those reported by Lee et al.[17] and Androulakis et al.[23] with a rms of 0.13 degrees. Those authors have consistently defined this conformation as that corresponding to the presumptive global energy minimum for Met-enkephalin.

Table III summarizes the computational requirements of several other methods that have addressed the same problem; i.e., they have considered all 24 dihedral angles of Met-enkephalin as variables, and have used ECEPP as the objective function. Based on this comparison, it is clear that TS compares favorably with the most popular methods.

Several years ago we experimented with the use of discrete variables in a heuristic search employing Threshold Accepting,[14] a Simulated Annealing variant. In that article we reported a lowest energy minimum conformation for Met-enkephalin (24 variables, Arms = 4.74) employing ECEPP/2 as the cost function. This structure was minimized with SUMSL[43] (TAO column in Table II), and the resulting dihedral angles are in excellent agreement (Arms = 0.08) with those reported by Androulakis et al.[23] In contrast, the structure reported by Wang and Pachter,[44] who employed ECEPP/2 as the potential function (Lrms = 82.1), a reduced set of 19 variables ($\omega$ angles fixed to 180°) and a continuous autoadaptive simulated annealing did not converged after minimization (ASAO column in Table II, Lrms = 78.3) to any of the structures

**TABLE II.**
**Dihedral Angles for Six Best Low-Energy Conformations for Met-Enkephalin.[a]**

| | | MCM | $\alpha$BB[b] | CSA[b] | **TS** | TSO | TAO | ASAO |
|---|---|---|---|---|---|---|---|---|
| Tyr | $\phi$ | −86.2 | −83.5 | −83.5 | **−86.5** | −83.3 | −83.7 | −84.4 |
| | $\psi$ | 156.2 | 155.8 | 155.8 | **151.0** | 155.7 | 155.8 | 151.9 |
| | $\omega$ | −176.9 | −177.1 | −177.2 | **−172.0** | −177.1 | −177.1 | −179.5 |
| | $\chi_1$ | −172.6 | −173.2 | −173.2 | **−175.0** | −173.2 | −173.2 | 178.8 |
| | $\chi_2$ | 78.7 | −100.5 | 79.4 | **−99.0** | −100.7 | −100.6 | −111.3 |
| | $\chi_3$ | −166.0 | 13.6 | −166.4 | **15.0** | 13.7 | 13.6 | 146.4 |
| Gly | $\phi$ | −154.5 | −154.3 | −154.3 | **−159.0** | −154.2 | −154.3 | −159.6 |
| | $\psi$ | 83.6 | 86.0 | 86.0 | **78.0** | 85.8 | 85.9 | 70.6 |
| | $\omega$ | 168.6 | 168.5 | 168.5 | **170.5** | 168.5 | 168.5 | 177.7 |
| Gly | $\phi$ | 83.7 | 82.9 | 82.9 | **84.0** | 82.9 | 82.9 | 66.8 |
| | $\psi$ | −73.8 | −75.1 | −75.1 | **−72.0** | −75.0 | −75.1 | −94.6 |
| | $\omega$ | −170.2 | −169.9 | −170.0 | **−170.0** | −169.9 | −169.9 | 177.7 |
| Phe | $\phi$ | −137.0 | −136.9 | −136.9 | **−139.0** | −136.8 | −136.8 | −81.6 |
| | $\psi$ | 19.3 | 19.1 | 19.1 | **23.0** | 19.1 | 19.1 | −26.9 |
| | $\omega$ | −174.1 | −174.1 | −174.1 | **−178.0** | −174.1 | −174.1 | −179.2 |
| | $\chi_1$ | 58.7 | 58.8 | 58.9 | **55.0** | 58.8 | 58.9 | 72.6 |
| | $\chi_2$ | −85.5 | −85.5 | 94.6 | **93.0** | 94.5 | −85.5 | 84.7 |
| Met | $\phi$ | −163.6 | −163.5 | −163.5 | **−165.0** | −163.4 | −163.4 | −78.1 |
| | $\psi$ | 160.4 | 160.9 | 161.2 | **161.0** | 160.8 | 160.8 | 131.6 |
| | $\omega$ | −179.7 | −179.8 | −179.8 | **178.0** | −179.8 | −179.8 | −179.1 |
| | $\chi_1$ | 52.7 | 52.9 | 52.9 | **54.0** | 52.8 | 52.9 | −171.3 |
| | $\chi_2$ | 175.2 | 175.3 | 175.3 | **−178.0** | 175.3 | 175.3 | 176.6 |
| | $\chi_3$ | −179.9 | −179.8 | −179.8 | **175.0** | −179.8 | −179.9 | 179.5 |
| | $\chi_4$ | −58.5 | 61.4 | −58.6 | **59.0** | 61.4 | 61.4 | −60.1 |
| Energy kcal/mol | | −12.371 | −12.389 | −12.389 | **−11.757** | −12.389 | −12.389 | −10.802 |

MCM—Nayeem, A.; Vila, J.; Scheraga, H. A. J Comp Chem 1991, 12, 594.
$\alpha$BB—Androulakis, I. P.; Maranas, C. D.; Floudas, C. A. J Global Opt 1997, 11, 1.
CSA—Lee, J.; Scheraga, H. A.; Rackovsky, S. J Comp Chem 1997, 18, 1222.
TS—This work.
TSO—After minimization of the best geometry from TS.
TAO—After minimization of the structure reported by Morales, L. B.; Garduño–Juárez, R.; Romero, D. J Biomol Struct Dynam 1992, 9, 951.
ASAO—After minimization of the structure reported by Wang, Z.; Pachter, R. J Comp Chem 1997, 18, 323.
[a] The tabulated energy values were computed with the most current version of ECEPP/3.
[b] Originaly the authors reported an energy value of −11.707 kcal/mol due to the slight differences in the ECEPP/3 parameters used in the computations cited.

shown in Table II. The fact that we have consistently obtained good lowest conformational energy structures for Met-enkephalin based on a heuristic search in a discrete variable space encourages us to try this method on longer peptides.

The best performance of the parallel TS algorithm on the SGI Origin 2000 was obtained when it was compiled without option −O2. The scale-up factor after parallelization of our algorithm is depicted in Figure 2. The speed-up factor shown in Figure 2 clearly shows the advantage of parallel design of a tabu search. For example, when we used 16 processors we obtained a speed-up of 12.38.

We conclude that TS is well suited for conformational searches of peptides, even if one works with discrete variables. It is obvious that it is always possible to improve the performance of the heuristic. To achieve this, one has to experiment with different values for the tabu parameters, such as the tabu distance, the length of the tabu list, the length of the long-term memory, the frequency criterion, and possibly to fine tune the diversification and the in-

**TABLE III.**
**Comparison of TS with Other Approaches for the Global Minimum Search of Met-Enkephalin Using ECEPP.**

| Method | $N_{var}$ | CPU hr | Function Evaluations $10^5$ | Computer | Mflops[a] |
|---|---|---|---|---|---|
| Monte Carlo minimization[12a, 12b] | 19[b] | 2–3 | 1.0 | IBM 3090 | 7.5 |
| | 24 | 10 | — | | |
| Simulated annealing[13] | 24 | 2.5 | 2.5 | Apollo DN10000 | 5.8 |
| Threshold accepting[14] | 24 | 1.5 | 2.0 | Apollo DN10000 | 5.8 |
| Monte Carlo with minimization[15a] | 24 | 1.5–4 | — | IBM 3090 | 7.5 |
| Multicanonical algorithm[16] | 19[b] | 6 | 1.5 | IBM RS/6000 320H | 12 |
| Conformational space annealing[17] | 24 | 0.75 | 1.7 | SG Indigo 2 | 32 |
| Diffusion equation[21a, 21b] | 19[b] | 0.33 | — | IBM 3090 | 7.5 |
| Mean field theory[22] | 10[b] | 1.6 | — | IBM 3090 | 7.5 |
| $\alpha$BB[23] | 24 | 1.3 | 3.9 | HP/9000 730 | 24 |
| This work | 24 | 0.07 | 1.7 | SG Origin 2000 | 114 |

[a] Jack Dongarra, http://performance.netlib.org/performance/html/linpack.data.co10.html.
[b] With these number of variables the corresponding methods arrived to an apparent global minimum.

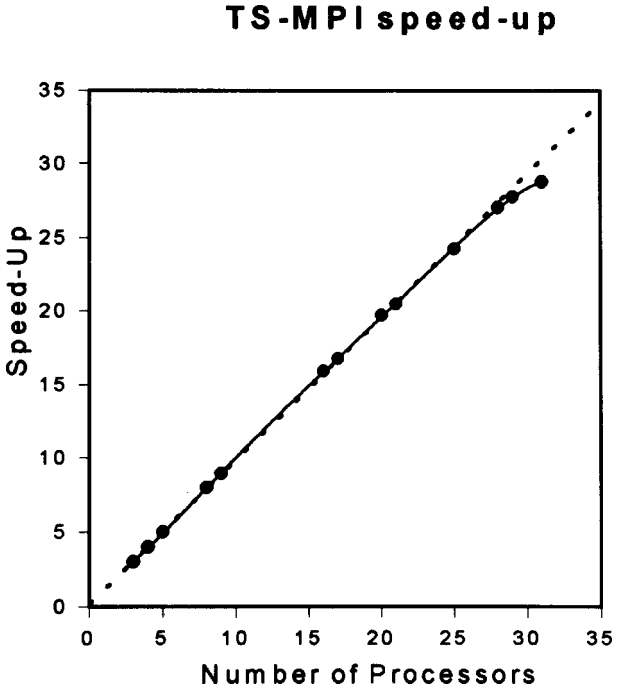


**FIGURE 2.** Ideal Slope up 25 processors. Actual Working Slope up to 32 processors.

tensification processes. Application of TS for larger peptides is under way.

## Acknowledgments

Two of the authors, A.A.J.M. and R.C.F.J. thank CRAY-UNAM and PAPIIT-DGAPA-UNAM for their fellowships. Dirección General de Servicios de Cómputo Académico at UNAM generously provided the computer time used. We thank Dr. Harvey Bialy Bloom for helpful comments.

## References

1. Anfisen, C. B. Science 1973, 181, 223.
2. Momany, F. A.; McGuire, R. F.; Burgess, A. W.; Scheraga, H. A. J Phys Chem 1975, 79, 2361.
3. Allinger, N. L. J Am Chem Soc 1977, 99, 8127.
4. Némethy, N.; Pottle, M. S.; Scheraga, H. A. J Phys Chem 1983, 89, 1883.
5. Brooks, B.; Bruccoleri, R.; Olafson, B.; States, S.; Swaminathan, S.; Karplus, M. J Comp Chem 1983, 4, 187.
6. Dauber–Osguthorpe, P.; Roberts, V. A.; Osguthorpe, D. J.; Wolff, J.; Genest, M.; Hagler, A. T. Proteins Struct Funct Genet 1988, 4, 31.
7. Weiner, S.; Kollmann, P.; Case, D. A.; Singh, U. C.; Ghio, C.; Alagona, G.; Profeta, S.; Weiner, P. J Am Chem Soc 1984, 106, 765.
8. van Gunsteren, W. F.; Berendsen, H. J. C. GROMOS; Groningen Molecular Simulation: Groningen, The Netherlands, 1987.
9. Allinger, N. L.; Yuh, Y. H.; Lii, J. I. J Am Chem Soc 1989, 111, 8551.
10. Némethy, G.; Gibson, K. D.; Palmer, K. A.; Yoon, C. N.; Paterlini, G.; Zagari, A.; Rumsey, S.; Scheraga, H. A. J Phys Chem 1992, 96, 6472.
11. Hart, W. E.; Istrail, S. J Comp Biol 1997, 4, 1.
12. (a) Li, Z.; Scheraga, H. A. Proc Natl Acad Sci USA 1987, 84, 6611; (b) Li, Z.; Scheraga, H. A. J Mol Struct (Theochem) 1988, 179, 333.

13. Morales, L. B.; Garduño–Juárez, R.; Romero, D. J Biomol Struct Dynam 1991, 8, 721.
14. Morales, L. B.; Garduño–Juárez, R.; Romero, R. J Biomol Struct Dynam 1992, 9, 951.
15. (a) Nayeem, G. A.; Vila, J.; Scheraga, H. A. J Comput Chem 1991, 12, 594; (b) Vásquez, M.; Meirovitch, E.; Meirovitch, H. J Phys Chem 1994, 98, 9380.
16. Hansmann, U. H. E.; Okamoto, Y. J Comp Chem 1993, 14, 1333.
17. Lee, J.; Scheraga, H. A.; Rackovsky, S. J Comp Chem 1997, 18, 1222.
18. Jin, A. Y.; Leung, F. Y.; Weaver, D. F. J Comp Chem 1997, 16, 1971.
19. Gibson, K. D.; Scheraga, H. A. J Comp Chem 1990, 11, 468.
20. Vajda, S.; DeLisi, C. Biololymers 1990, 129, 1755.
21. (a) Kostrowicki, J.; Piela, L.; Cherayil, B. J.; Scheraga, H. A. J Phys Chem 1991, 95, 4113; (b) Kostrowicki, J.; Scheraga, H. A. J Phys Chem 1992, 96, 7442.
22. Olszewski, K. A.; Piela, L.; Scheraga, H. A. J Phys Chem 1992, 96, 4672.
23. Androulakis, I. P.; Maranas, C. D.; Floudas, C. A. J Global Opt 1997, 11, 1.
24. Némethy, G.; Pottle, M. S.; Scheraga, H. A. J Phys Chem 1983, 87, 1883.
25. Glover, F. ORSA J Comput 1989, 1, 190.
26. Garcia, B.; Toulouse, M. Comput Oper Res 1994, 21, 1025.
27. Fiechter, C. N. Discrete Appl Math 1994, 51, 243.
28. Taillard, E. ORSA J Comput 1994, 6, 108.
29. Morales, L. B. Math Magazine 1997, 70, 287.
30. Kvasnicka, V.; Pospíchal, J. J Chem Inf Comput Sci 1994, 34, 1109.
31. Pardalos, P. M.; Liu, X.; Xue, G. L. J Global Opt 1997, 11, 55.
32. Cvijovic, D.; Klinowski, J. Science 1995, 267, 664.
33. Battiti, R.; Tecchiolli, G. Ann Oper Res 1996, 63, 153.
34. Westhead, D. R.; Clark, D. E.; Murray, C. W. J Comput Aided Mol Design 1997, 11, 209.
35. Laguna, M.; Glover, G. Manage Sci 1993, 39, 492.
36. Chakrapani, J.; Skorin–Kapov, J. Methods Oper Res 1993, 36, 150.
37. Levitt, M. J Mol Biol 1983, 168, 621.
38. Post, C. B.; Dobson, C. M.; Karplus, M. Proteins Struct Funct Genet 1989, 5, 337.
39. Taillard, E. Parallel Comput 1991, 7, 443.
40. Foster, F. Designing and Building Parallel Programs; Addison–Wesley: Reading, MA, 1995.
41. Walker, D. W. Parallel Comput 1994, 20, 657.
42. Feldman, S. I.; Gay, D. M.; Maimone, M. W.; Schryer, N. L. A Fortran to C Converter; Computing Science Technical Report No. 149, 1995, AT & T Bell Laboratories, Murray Hill, NJ 07974.
43. Gay, D. M. ACM Trans Math Software 1983, 9, 503.
44. Wang, Z.; Pachter, R. J Comp Chem 1997, 18, 323.